REMARKS/ARGUMENTS

These remarks are submitted responsive to the Office Action of July 13, 2005 (hereafter Office Action). As this response is timely filed within the 3-month shortened statutory period, no fee is believed due.

In paragraphs 11-12, Claims 1-6, 11, and 16-21 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication No. 2003/0069908 to Anthony, *et al.* (hereinafter Anthony) in view of "Understanding UML: The Developer's Guide With a Web-Based Application in Java", by Fowler, *et al.* (hereinafter Fowler), and in further view of U.S. Patent Publication No. 2002/0016828 to Daugherty, *et al.*, (hereinafter Daugherty). In paragraph 13, Claims 7-8 and 12-15 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Anthony, in view of Fowler and in further view of "Laura Lemay's Web Workshop JavaScript", by Lemay, *et al.* (hereinafter Lemay).

Applicant has amended independent Claims 1, 7, 11, and 16 to emphasize certain aspects of Applicant's invention. The amendments are supported throughout the Specification. (See, e.g., Specification, p. 3, lines 18-23 and lines 28-29; p. 4, lines 9-10; p. 6, lines 13-14; and p. 7, lines 2-4.) No new matter has been introduced by virtue of the claim amendments.

I. Applicant's Invention

It may be useful to reiterate certain aspects of Applicant's invention prior to addressing the cited references. The invention provides a mechanism whereby state chart data produced by conventional modeling tools can be transformed into markup language representations. (See, e.g., Specification, p. 3, lines 1-10.) With the invention, a modeling language representation generated by a state-machine modeling tool is converted into an XML document that represents a modeling language-compliant state-machine model.

One embodiment of the invention, typified by amended independent Claim 1, is a method for defining a markup language representation of state chart data. The method includes loading state chart data corresponding to a state chart diagram through an interface to a state machine modeling tool. The state chart diagram specifies the behavior of a plurality of objects. The state chart data specifies the life-cycle states possible for each object and the behavior of the objects in each specified state. The state chart data is itself specified according to a modeling language. The modeling language can be the uniform modeling language, UML, or any other modeling language. (Specification, p. 7, lines 2-7.)

The method further includes generating header data in accordance with a selected markup language and retrieving a state name and state transition data from the state chart data for each state specified in the state chart data. The state transition data specifies event occurrences for transitioning from one state to another. The method then proceeds to format the retrieved state names and corresponding state transition data. In particular, the step results in the state names and corresponding state transition data being formatted according to the selected markup language. The header data along with the formatted state names and state transition data is then saved in a document formatted according to the selected markup language.

II. The Claims Define Over The Prior Art

As already noted, independent Claims 1, 11, and 16 were rejected as unpatentable over Anthony in view of Fowler and Daugherty. Applicant respectfully submits, however, that the combination fails to teach or suggest every feature recited in the claims and that the prior art fails to provide a suggestion, teaching, or motivation for combining the references.

Fowler discloses what is already widely known, namely, that a Unified Modeling Language (UML) state diagram can be used to conveniently and succinctly describe the

behavior of a system. Anthony, as noted at page 4 of the Office Action, translates one XML document into another XML document. Yet state diagrams and XML documents are wholly disparate concepts, and what neither reference discloses is how to transform state chart data into a markup language representation. Specifically, neither reference, alone or in combination, teaches or suggests a mechanism for transforming state chart data specified according to a modeling language into a markup language representation, as recited in amended independent Claims 1, 11, and 16.

At page 4 of the Office Action, it is stated that because Anthony teaches the translation of documents in one XML format into documents in a different XML format, it would have been obvious to combine Anthony with Fowler in order to provide the benefit of "smoothly and efficiently exchanging information among users who implement different XML document formats." What neither reference answers, though, is how to put Fowler's UML-standard representation into an XML format so that it can be translated by Anthony into a different XML format.

As explicitly noted in the Office Action, Anthony's translation from one XML format to another is based on using the document type definition (DTD) of a first XML document to translate the document into a different XML document. The UML, however, provides no such DTD. The UML is a standard specification for modeling system concepts using, for example, class diagrams and state machine diagrams. Such diagrams as provided by the UML have no relationship to DTDs specifically or to XMLs generally.

The XML translation taught by Anthony, as noted above and in the Office Action, starts with a specific source DTD corresponding to the source XML document (a text document) and translates the source DTD to a target DTD in order create a target XML document (another text document). (Paragraphs 0009 and 0011; Abstract.) Yet the UML does not provide nor depend on DTDs. One of ordinary skill, looking at the UML state-machine diagram in Fowler and the XML-to-XML translation of Anthony, is left with no

teaching as to how to use Anthony to transform the UML state-machine diagram of Fowler – which lacks the DTD that Anthony depends on for XML-to-XML translations – into an XML-formatted document.

The UML of Fowler provides the advantage of a succinct and efficient modeling of system behavior. The XML-to-XML translation of Anthony provides for efficient exchange of information among users of differently-formatted XML documents. Yet the prior art provides no teaching or suggestion for combining the references, since the prior art does not teach or suggest how to transform Fowler's UML into an XML document amenable to Anthony's XML-to-XML translation. Anthony and Fowler are, in fact, directed to different problems, the former to XML translations and the latter to system modeling. Anthony's reliance, for example, on DTDs which Fowler's UML does not provide precludes even an implied motivation to combine the references.

More fundamentally, since neither Anthony nor Fowler provides the mechanism by which data specified according to a modeling language can be transformed into an XML format, the combination fails to produce Applicant's invention. For example, neither of the references teaches or suggests retrieving state names with corresponding state transition data from state chart data, specified according to a modeling language, and then formatting the state names and corresponding state transition data according to a selected markup language, as recited in amended independent Claims 1 and 16. Similarly, neither reference teaches or suggests, for example, a markup language formatter for formatting in a markup language the state chart data and component state actions specified according to a modeling language, as recited in amended independent Claim 11.

Daugherty is cited at page 5 of the Office Action as disclosing the generation of header data in accordance with a selected markup language. Like Anthony and Fowler, however, Daugherty does not teach or suggest anything remotely similar to the features recited in amended independent Claims 1, 11, and 16. Specifically, Daugherty does not

remotely teach or suggest transforming data specified according to a modeling language into an XML format.

With respect to independent Claim 7, Anthony is cited at page 10 of the Office Action as teaching the formatting of state chart data according to a selected markup language, and Fowler is cited as teaching the generation of a state diagram. Lemay is cited as teaching the "combining of a JavaScript program with an HTML document for performing certain functions."

Applicant respectfully reasserts that neither Anthony's XML-to-XML translations nor Fowler's UML state diagrams remotely suggest, for example, formatting state chart data, initially specified into a modeling language, into a markup language representation according to a selected markup language, as recited in amended independent Claim 7. Anthony translates a document in one XML format into another, but even when combined with Fowler, suggests no mechanism for transforming state chart data specified according to a modeling language into a markup language representation. Fowler describes the attributes of the UML, but provides no mechanism for transforming UML representations into XML representations.

Claim 7 expressly recites an add-in script that is used in conjunction with a state machine tool, which generates state chart data specified according to a modeling language. The add-in script formats the state chart data specified according to a modeling language into a markup language representation. Lemay merely describes extending an HTML document using JavaScript. The example Lemay describes is using JavaScript to create a Web page that displays a particular greeting depending on the time of day. (pp. 8-9.) The application of JavaScript in the context of HTML documents in order to produce "a relaxed program environment," however, suggests nothing about an add-in script that transforms modeling language state chart data into an XML representation, as taught by Applicant's invention. In particular, the extension of HTML programming using JavaScript does not teach or suggest an add-in script that formats

state chart data specified according to a modeling language into a markup language representation, as recited in amended independent Claim 7.

With respect to independent Claim 14, Anthony is cited as disclosing generating a model of an XML document using a modeling system. Fowler is cited as teaching generating a UML state diagram having state names and transition data. At page 13 of the Office Action, it is noted that neither reference discloses an add-in script defining a markup representation of a UML-specified state chart diagram or a state-machine run-time engine executing the markup language representation. Lemay is cited as disclosing both of the latter features.

As already noted, Lemay explicitly describes extending HTML documents using JavaScript. Nothing in Lemay's description of JavaScript, however, remotely suggests an add-in script that leads to a markup language representation of a UML-specified state chart diagram produced by a state-machine modeling tool, as recited in independent Claim 14. Lemay's JavaScript begins and ends with an HTML document. In this respect, Lemay fails to suggest Applicant's invention for the same reason Anthony does. An HTML document is not a UML-specified state chart diagram, and Lemay provides no mechanism to transform the latter into the former. With Applicant's invention, the UML-specified state chart document is transformed into a markup language representation by an add-in script. Lemay's JavaScript does not provide this capability. Even when combined, therefore, Anthony, Fowler, and Lemay fail to teach or suggest every feature of independent Claim 7.

Applicant respectfully submits that none of the references, alone or in combination, teach or suggest every feature recited in independent Claims 1, 7, 11, and 16, as amended. None of the references alone or in combination teach or suggest every feature recited in independent Claim 14. Applicant, therefore, respectfully maintains that the independent claims each define over the prior art. Applicant further respectfully submits that whereas the remaining claims each depend from one of the independent

claims while reciting additional features, the dependent claims likewise define over the prior art.

CONCLUSION

Applicant believes that this application is now in full condition for allowance, which action is respectfully requested. The Applicant requests that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the subject application to completion.

Respectfully submitted,

Date: October 13, 2005

Gregory A. Nelson, Registration No. 30,577
Richard A. Hinson, Registration No. 47,652
Marc A. Boillot, Registration No. 56,164
AKERMAN SENTERFITT
Customer No. 40987
Post Office Box 3188
West Palm Beach, FL 33402-3188
Telephone: (561) 653-5000